AI-Based Conference Program Scheduler

AIware Challenge 2024

Rucha Deshpande
Computer Science
Purdue University
West Lafayette, Indiana, USA
deshpa37@purdue.edu

Aishwarya Devi Akila Pandian
Computer Science
Purdue University
West Lafayette, Indiana, USA
aakilapa@purdue.edu

Vigneshwaran Dharmalingam
Computer Science
Purdue University
West Lafayette, Indiana, USA
vdharmal@purdue.edu

ABSTRACT

Program creation is the process of allocating presentation slots for each paper accepted to a conference with parallel sessions. This process, generally done manually, involves intricate decision-making to align with multiple constraints and optimize goals. We propose an AI-based Conference Program Scheduler, a tool that uses AI-based techniques to automate the conference schedule creation process while addressing given constraints and maximizing session theme coherence. This project is motivated by the necessity to streamline the conference scheduling process, aiming to reduce the substantial manual effort currently required by program committee chairs[1]. We plan to use machine learning techniques like clustering for grouping papers on similar topics and optimization algorithms like genetic algorithm to allocate sessions effectively. We plan to compare our results with the previous conference schedule data to evaluate the effectiveness and efficiency of the AI-based Conference Program Scheduler. Our evaluation will focus on the tool's ability to meet the outlined constraints and its success in creating thematically coherent sessions using metrics like constraint violation count and thematic coherence score.

CCS CONCEPTS

• Computing methodologies → Artificial intelligence, Genetic algorithms, Natural language processing

• Theory of computation → Evolutionary algorithms

KEYWORDS

Automated Scheduling, Genetic Algorithms, Optimization Algorithms, Semantic Analysis

ACM Reference format:

Rucha Deshpande, Aishwarya Devi Akila Pandian and Vigneshwaran Dharmalingam. 2024. AI-Based Conference Program Scheduler: AIware Challenge 2024.

1 INTRODUCTION

In the realm of academic and professional conferences, scheduling paper presentations is an important and complex task. Program Chairs have to manually allocate presentation slots while taking care of different constraints, including thematic alignment, speaker availability, and audience interest. Traditional manual scheduling methods are time-consuming, error-prone, and often fail to optimize for participant engagement. This can lead to suboptimal session timings, thematic mismatches, and logistical conflicts. The need for an automated scheduling tool arises from the requirement to overcome the limitations of manual scheduling. We propose an AI-based Conference Program Scheduler to automate the conference schedule creation process while addressing given constraints. The constraints that make the creation of a program challenging are:

1. Total time for all presentations in a session cannot be longer than the length of the session.
2. Total number of sessions has to be equal to the number of sessions provided by the Program Committee chairs.
3. If there are parallel tracks, then no two papers with common authors can be scheduled in parallel at the same time.

Another important consideration is the optimization goal: the papers in a session should be on a similar topic to avoid the parallel scheduling of sessions with related topics. The goal of our project is to reduce the manual effort required and minimize errors by automating this program creation process. Our proposed solution is based on machine learning techniques for semantic analysis and optimization algorithms for schedule construction. Semantic Analysis involves utilizing state-of-the-art text embedding techniques, such as Doc2Vec or BERT, to transform the textual content of submissions (titles, abstracts, and keywords) into high-dimensional vectors that capture semantic meaning, and further applying clustering algorithms on the embeddings to group papers into coherent sessions. This step is crucial for ensuring that sessions are topic-wise aligned.

After obtaining the topic-wise grouped papers, the next phase involves constructing the actual conference schedule, which adheres to the provided constraints (session lengths, number of sessions, parallel track limitations). Most scheduling problems, especially those with multiple constraints and optimization criteria, fall into the category of NP-hard problems. Adaptive heuristic algorithms such as the Genetic Algorithm[2] are suitable for this type of scheduling problem as they often find good solutions in a reasonable amount of time. Genetic Algorithms are a type of evolutionary algorithm and optimization technique inspired by the process of natural selection. They iterate over generations of solutions to find the best or most fit solution according to a defined fitness function. Genetic Algorithms can optimize the overall schedule considering constraints and objectives[3]. A related study by Bassen et al. (2020)[5] investigates the application of reinforcement learning for dynamically scheduling educational activities in extensive online courses. Substituting RL with Genetic Algorithms could potentially mitigate several challenges highlighted in the research. One of the limitations mentioned is the significant amount of data required for the reinforcement learning model to converge to an effective solution. GAs require less data to find a satisfactory solution because they can explore a wide solution space more broadly and might discover good solutions with less interaction data. Moreover, the adaptive nature of GAs allows for periodic adjustments to the scheduling policy. Hence, we plan to apply the Genetic Algorithm for our AI-based Conference Program Scheduler.

Applying the Genetic Algorithm to the conference scheduling problem involves several key steps:

1. **Representation (Encoding):** Each potential conference schedule will be represented as an individual in the population. The schedule will be encoded as a matrix, where each element represents a session slot and contains information about the paper(s) scheduled in that slot. Each gene within an individual will represent a specific aspect of the schedule, such as the assignment of a particular paper to a session and track.
2. **Fitness Function:** The fitness function will evaluate how well a given schedule meets the objectives and constraints. This will include penalties for constraint violations (e.g., scheduling overlaps for authors) and rewards for desired features (e.g., thematic coherence within sessions).
3. **Selection:** Tournament selection or roulette wheel selection will be used, favoring individuals with higher fitness scores.
4. **Crossover:** Session slots will be swapped between schedules, ensuring that the resulting offspring still respects the constraints of the problem.
5. **Mutation:** Mutation will involve moving a paper to a different session, again ensuring that any changes do not violate the problem's constraints.
6. **Evaluation and selection of the next generation:** The fitness function will be used to evaluate the individual schedules and decide which ones to carry forward for the next generation.
7. **Termination:** This could be a set number of generations, a fitness threshold that indicates an acceptable quality of schedule, or a lack of significant improvement over multiple generations.

To validate the effectiveness of our tool, we plan to conduct an evaluation using real-world conference data from the MSR 2022[6] and MSR 2023[7] programs. This will involve testing the tool's ability to meet the scheduling constraints outlined by the challenge and its success in creating thematically coherent sessions. We plan to use metrics such as constraint violation count and thematic coherence score (the average similarity of topics within sessions). We will compare the schedules generated by our tool against those created manually in terms of thematic coherence and adherence to constraints, to assess the improvements offered by our solution.

We are also exploring LLM-based techniques to enhance various stages of our Genetic Algorithm-based approach. LMEA[4] or LLM-driven Evolutionary Algorithms is a promising direction to explore. In each generation of the evolutionary search, LMEA constructs a prompt to instruct the LLM to select parent solutions from the current population and perform crossover and mutation to generate offspring solutions. Then, these new solutions are evaluated and added to the population for the next generation. Due to the capabilities of LLMs, we can describe the optimization problem and the desired solution properties in natural language to instruct the LLM. In consequence, optimization with LMEA enables quick adaptation to different optimization problems. The authors of LMEA have experimented with the classical traveling salesman problem. However, we still have to understand how this approach will work in complex problems such as scheduling with constraints.

REFERENCES

[1] AIware 2024 - challenge track - aiware 2024 (no date) AIware 2024 - Challenge Track - AIware 2024. Available at: https://2024.aiwareconf.org/track/aiware-2024-challenge-track (Accessed: 18 February 2024).
[2] Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73.
[3] Burke, E. K., Elliman, D., & Weare, R. (1994, September). A genetic algorithm based university timetabling system. In Proceedings of the 2nd east-west international conference on computer technologies in education (Vol. 1, pp. 35-40).
[4] Liu, S., Chen, C., Qu, X., Tang, K., & Ong, Y. S. (2023). Large language models as evolutionary optimizers. arXiv preprint arXiv:2310.19046.
[5] Jonathan, B., Bharathan, B., Michael, S., Candace T., Jay P., Dawn Z., Alex G., Ethan F, John C M.. (2020). Reinforcement Learning for the Adaptive Scheduling of Educational Activities. CHI 2020.
[6]https://conf.researchr.org/program/msr-2022/program-msr-2022/Detailed-Table
[7]https://conf.researchr.org/program/msr-2023/program-msr-2023/Detailed-Table